WHAT IS CLAIMED IS

1. An input device control program which allows multiple client application programs to simultaneously communicate with a single input device,

wherein said input device control program is loaded as a process, and

wherein all subsequent application programs call to said process to establish communication with said single input device.

2. The input device control program of claim 1 wherein said input device comprises a digital Internet video camera.

3. The input device control program of claim 1 wherein said input device comprises a microphone.

4. The input device control program of claim 1 wherein said input device control program comprises routines for:

a) video control methods comprising:

i) initializing a video control;

ii) taking digital still images;

iii) recording digital video images;

iv) obtaining video driver information;

v) setting video camera properties; and

vi) obtaining video camera properties;

b) video camera event notification comprising:

i) motion detection notification;

ii) audio visual (AVI) error notification;

iii) camera detached notification; and

iv) camera reattached notification.

5. The input device control program of claim 1 wherein said process handles all details of network protocols comprising:

loading said input device control program;

calling to said input device control program with relevant input/output data;

buffering input and output to/from said input device control program;

executing said input device control program; and

unloading said input device control program.

6. A input device control program which allows multiple client application programs to simultaneously communicate with an input device, wherein said input device control program in response to a first application program calling for a first connection to be established to said input device:

i) passes said first application program's calls to a process' application program interface (API);

ii) causes said process' network protocol to load said executable input device control program onto a process server;

iii) causes said process server to create a single input device instance and connects said single input device instance to said input device;

iv) causes said process server to create a first input device control instance and connects said first input device control instance to said single input device instance;

v) causes said process server to create an interface through which said client application program communicates with said single input device instance, and

vi) causes a second input device control instance to be created in response to a call from a second application program calling for a second connection to a said single input device and connecting said second input device control instance to said single input device instance allowing said second application program to communicate with said same single input device instance.

7. The input device program of claim 6 wherein said input device control program is a distributed component object model (DCOM) executable program which comprises routines for:

a) video control methods comprising:

i) initializing a video control;

ii) taking digital still images;

iii) recording digital video images;

iv) obtaining video driver information;

v) setting video camera properties; and

vi) obtaining video camera properties;

b) video camera event notification comprising:

i) motion detection notification;

ii) audiovisual AVI error notification;

iii) camera detached notification; and

iv) camera reattached notification.

8. The input device control program of claim 6 wherein said process is a distributed component object model (DCOM) executable program.

9. A distributed component object model (DCOM) executable input device control program which allows multiple client application programs to simultaneously communicate with a input device, wherein said program in response to a first application program calling for a first connection to be established to said input device:

i) passes said first application program's calls to a DCOM application program interface (API);

ii) causes said DCOM's network protocol to load said executable input device control program onto a DCOM server and;

iii) causes said DCOM server to create a single input device instance and connects said single input device instance to said input device;

iv) causes said DCOM server to create a first input device control instance and connects said first input device control instance to said single input device instance;

v) causes said DCOM server to create an interface through which said;

client application program communicates with said single input device instance, and

vi) creates a second input device control instance to be created in response to a call from a second application program calling for a second connection to a said single input device and connecting said input device instance to said single input device instance allowing said second application program to communicate with said same single input device instance.

10. A computer useable medium having computer readable code embodied therein for causing the simultaneous sharing of an input device by multiple application programs running on a host and calling to said input device,

said computer readable code virtualizing an input device driver file,

where said computer readable code is implemented in an executable client-sever architecture, where each of said application programs is a client, and

where said computer readable code is a server.

11. The computer useable medium of claim 10 wherein said input device comprises a digital video camera interfaced with said host.

12. The computer useable medium of claim 10 wherein said input device comprises a microphone interfaced with said host.

13. The computer useable medium of claim 10 wherein said host is selected from the group consisting of a personal computer, a handheld computer, an interactive set-top box, a thin client computing device, a personal access device, a personal digital assistant, an internet appliance, an internet connected digital picture frame and combinations thereof.

14. The computer useable medium of claim 10 wherein said application programs communicate with said client-server architecture via a client side mechanism implemented as an input device portal.

15. The computer user medium of claim 14 wherein said input device portal is an ActiveX control.

16. The computer useable medium of claim 10 wherein said application programs communicate with said client-server architecture via a client side mechanism implemented as a virtual source filter.

17. A method for allowing multiple client application programs to communicate with a single input device, said method comprising:

i) passing a first application program's calls to a process' application program interface (API);

ii) causing said process' network protocol to load an executable input device control program onto a process server and;

iii) causing said process server to create a single input device instance and connecting said single input device instance to said single input device;

iv) causing said process server to create a first input device control instance and connecting said first input device control instance to said single input device instance;

v) causing said process server to create an interface through which said client application program communicates with said single input device instance, and

vi) creating a second input device control instance in response to a call from a second application program calling for a second connection to said single input device and connecting said second input device control instance to said single input device instance to create an interface through which said second client application communicates with said same single input device instance.

18. The method of claim 17 wherein said executable input device control program is a distributed component object model (DCOM) executable program.

19. The method of claim 17 wherein said process is a DCOM process.